

Long-Term Identity-Aware Multi-Person Tracking for Surveillance Video Summarization

Shoou-I Yu, Yi Yang, Xuanchong Li, and Alexander G. Hauptmann

Abstract—In multi-person tracking scenarios, gaining access to the identity of each tracked individual is crucial for many applications such as long-term surveillance video analysis. Therefore, we propose a long-term multi-person tracker which utilizes face recognition information to not only enhance tracking performance, but also assign identities to tracked people. As face recognition information is not available in many frames, the proposed tracker utilizes manifold learning techniques to propagate identity information to frames without face recognition information. Our tracker is formulated as a constrained quadratic optimization problem, which is solved with nonnegative matrix optimization techniques. Tracking experiments performed on challenging data sets, including a 116.25 hour complex indoor tracking data set, showed that our method is effective in tracking each individual. We further explored the utility of long-term identity-aware multi-person tracking output by performing video summarization experiments based on our tracking output. Results showed that the computed trajectories were sufficient to generate a reasonable visual diary (i.e. a summary of what a person did) for different people, thus potentially opening the door to summarization of hundreds or even thousands of hours of surveillance video.

Index Terms—Multi-Object Tracking, Nonnegative Matrix Optimization, Surveillance Video Summarization, Face Recognition

1 INTRODUCTION

SURVEILLANCE cameras have been widely deployed to enhance safety in our everyday lives. The recorded footage can further be used to analyze long term trends in the environment. Unfortunately, manual analysis of large amounts of surveillance video is very difficult, thus motivating the development of computational analysis of surveillance video. A common first step of computational analysis is to track each person in the scene, which has led to the development of many multi-object tracking algorithms [1], [2], [3]. However, two important points are largely neglected in the literature: 1) the usage of identity information such as face recognition, numbers on a sports jersey, or any other cue that can identify an individual, and 2) the exploration of real-world applications based on tracking output from hundreds or thousands of hours of surveillance video.

There are two main advantages of utilizing identity information for tracking. First, identity information such as face recognition empowers the tracking algorithm to relate a tracked person to a real-world living individual, thus enabling subsequent individual-specific activity analysis. Second, identity information can also enhance tracking performance as it pinpoints the location of a specific individual at a given time. This additional information reduces the chance of an identity-switch.

We propose a novel identity-aware tracking algorithm as follows. Under the tracking-by-detection framework [4], the tracking task can be viewed as assigning each person detection result to a specific individual. Label information for each person can be acquired from face recognition. However, as face recognition is not available in many frames, face recognition information is propagated to other frames using a manifold learning approach, which captures the appearance similarities and spatial-temporal layout of person detections. The manifold

learning approach is formulated as a constrained quadratic optimization problem and optimized with nonnegative matrix optimization techniques. The constraints included are the mutual exclusion and spatial locality constraints that constrain the final solution to deliver a reasonable multi-person tracking output.

We performed tracking experiments on challenging data sets, including a 116.25 hour complex indoor tracking data set. Our long-term tracking experiments show that our method is effective in localizing and tracking each individual in hundreds of hours of surveillance video. An example output of our algorithm is shown in Figure 1, which shows the location of each identified person on a map in the middle of the image. This is analogous to the *Marauder's Map* described in the Harry Potter book series [5].

To further explore the utility of multi-person tracking output from hundreds of hours of video, we performed *summarization-by-tracking* experiments based on tracking output to acquire the *visual diary* of a person. Visual diaries provide a person-specific summary from hundreds of hours of surveillance video by showing the snapshots and semantic textual descriptions of the activities performed by the person. An example is shown in Figure 2, where the visual diary of a nursing home resident is shown. Experiments conducted on 116.25 hours of video show that we are able to summarize surveillance video with reasonable accuracy, which further demonstrates the effectiveness of our tracker.

In sum, the main contributions of this paper are as follows:

- 1) We propose an identity-aware multi-object tracking algorithm. Our tracking algorithm leverages identity information which is utilized as sparse label information in a manifold learning framework. The algorithm is formulated as a constrained quadratic optimization problem and solved with nonnegative matrix optimization.
- 2) A 15-camera multi-object tracking data set consisting

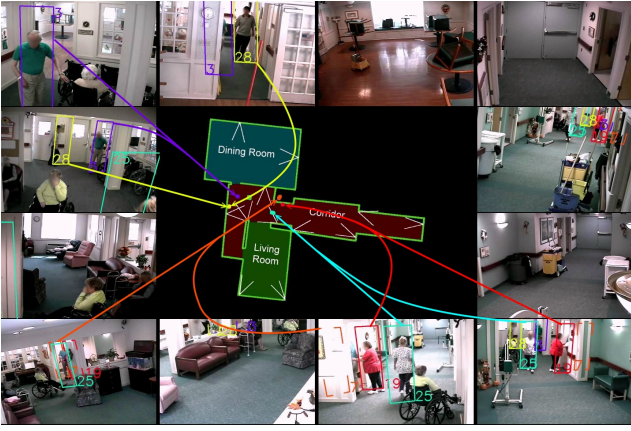


Fig. 1: The Marauder's Map for a nursing home (*Caremedia Short* sequence [6]) with the map in the middle. Dots on the map show the locations of different people. The surrounding images are the views from each surveillance camera. White lines correspond to the field-of-view of each camera.

of 116.25 hours of nursing home surveillance video was annotated. This real-world data set enables us to perform very long-term tracking experiments to better assess the performance and applicability of multi-object trackers.

- 3) Video summarization experiments based on tracking output were performed on 116.25 hours of video. We demonstrate that the visual diaries generated from tracking-based summarization can effectively summarize hundreds of hours of surveillance video.

The rest of the paper is organized as follows. Section 2 reviews related work on multi-object tracking. We detail our tracker in Section 3. Tracking and video summarization results are shown in Section 4, and Section 5 concludes the paper.

2 RELATED WORK

A main line of multi-object tracking work follows the tracking-by-detection paradigm [4], which has four main components: object localization, appearance modeling, motion modeling and data association. The object localization component generates a set of object location hypotheses for each frame. The localization hypotheses are usually noisy and contain false alarms and misdetections, so the task of the data association component is to robustly group the location hypotheses which belong to the same physical object to form many different object trajectories. The suitability of the grouping can be scored according to the coherence of the object's appearance and the smoothness of the object's motion, which correspond to appearance modeling and motion modeling respectively. We now describe the four components in more detail.

2.1 Object Localization

There are mainly three methods to find location hypotheses: using background subtraction, using object detectors, and connecting single-frame detection results into tracklets.

The Probabilistic Occupancy Map (POM, [7]) combines background subtraction information from multiple cameras to

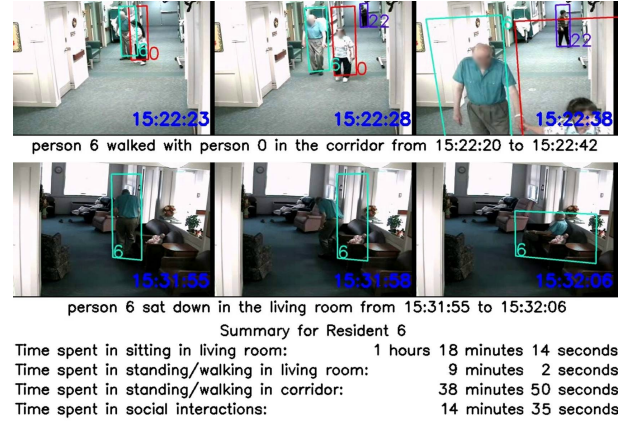


Fig. 2: An example visual diary for an elderly resident in a nursing home. The automatically generated textual description and three snapshots are shown for the two events. Long-term statistics are also shown.

jointly locate multiple objects in a single frame. It has been shown to be very effective in multi-camera environments [7], [8], [9], [10]. However, POM requires the discretization of the tracking space, and some precision may be lost. Also, when the placement of cameras is non-ideal, such as on long corridors where cameras only view the principal direction of the corridor [6], the POM localization results are not as accurate. Lastly, when there are different kinds of moving objects in the scene, POM cannot distinguish between the different kinds of tracked objects.

Utilizing object detector output is one of the most common ways to localize tracking targets [4], [11], [6], [12], [1], [13], [3], [14], [2], [15]. The main advantages of using object detectors are 1) enables the automatic initialization and termination of trajectories, and 2) alleviates template drift as the same detector is used for all frames. The main disadvantage is that a reliable general-purpose detector is required for the object to be tracked.

Localized objects in each frame could be connected to create *tracklets* [16], [17], [18], [19], [20], [21], [9], which are short tracks belonging to the same physical object. Tracklets are formed in a very conservative way to avoid connecting two physically different objects. As tracklets merge multiple location hypotheses, they can be used to enhance the efficiency of the tracking process [9].

2.2 Appearance Models

Appearance models discriminate between detections belonging to the same physical object and other objects. Color histograms [22], [23], [1], [16], [20], [24], [11], [6], [9] have been widely used to represent the appearance of objects, and the similarity of the histograms is often computed with the Bhattacharyya distance [1], [24]. Other features such as Histogram of Oriented Gradients [25] have also been used [17], [18].

Appearance models can also be learned from tracklets. The main assumption of tracklets is that all detections in a tracklet belong to the same object, and [17], [19], [21], [26], [27], [18] exploit this assumption to learn more discriminative

appearance models. Note that the “identity” in our work is different from [18], which utilized person re-identification techniques to improve the appearance model. We, however, focus on the “real-world identity” of the person, which is acquired from face-recognition.

In this work, we utilized color histograms combined with manifold learning to perform tracking. Manifold learning has also been utilized in previous work such as [28], [29] to learn subspaces for appearance features that can better differentiate the tracked target from other targets or background in single object or multi-object tracking settings. However, the multi-object tracking performed in [29] utilized multiple *independent* particle filters, which may have the issue of one particle filter “hijacking” the tracking target of another particle filter [30], [31]. Therefore, to fix this issue, our method has the mutual exclusion and spatial locality constraint encoded in the optimization framework, which *jointly* optimizes for all trajectories to acquire a potentially more reasonable set of trajectories.

2.3 Motion Models

Objects usually move in a smooth manner, and effective motion models can capture this assumption to better model the likely movement of objects. [1], [8], [12], [6], [9] use the bounded velocity model to model motion: given the current location of the object, the location in the next frame is constrained by the maximum velocity of the object. [23], [11], [15] improve upon this by modeling motion with the constant velocity model, which is able to model the smoothness of the object’s velocity change. Higher order methods such as spline-based methods [2], [3] and the Hankel matrix [32] can model even more sophisticated motions. [21] assumes that different objects in the same scene move in similar but potentially non-linear ways, and the motion of highly confident tracklets can be used to infer the motion of non-confident tracklets.

2.4 Data Association

A data association algorithm takes the object location hypotheses, appearance model and motion model as input to find a disjoint grouping of the object location hypotheses which best describes the motion of objects in the scene. Intuitively, the data association algorithm will decide whether to place two object location hypotheses in the same group based on their *affinity*, which is computed from the appearance and motion models.

For the association between multiple frames, there are two popular formulations: the Hungarian algorithm and the network flow, which are both Integer Linear Programs (ILP) with a special form. Given the pair-wise affinities, the Hungarian algorithm can find the optimal bipartite matching between two sets of object location hypotheses in polynomial time [16], [20], [18], [17], [2]. In the network flow formulation [1], [12], [8], [10], each path from source to sink corresponds to the trajectory of an object. The network flow problem can also be solved optimally in polynomial time, but this formulation and the Hungarian algorithm formulation makes a number of assumptions. The first assumption is that each physical object

can only be associated with one location hypothesis at each time instant to enforce the constraint that an object cannot be at multiple places at the same time. Therefore, in multi-camera environments, location hypotheses from multiple cameras need to be consolidated first with methods such as POM [7] before these methods can be used. The second assumption is that the cost function of each trajectory can be decomposed into a series of products (or additions) of pair-wise terms [2]. Therefore, most network flow-based methods are limited to the bounded velocity model, i.e. velocity from the previous time instant is not taken into account. [15], [24], [33] have improved on this by taking into account three location hypotheses at once, so the constant velocity model can be utilized. However, this comes at the cost of using more complex algorithms such as Lagrangian Relaxation [15] or using a Linear Program solver to approximately solve an ILP [24], where finding the global optimum is no longer guaranteed. Another method to incorporate velocity in such a framework is to utilize tracklets as the basic unit of location hypotheses [16].

Many trackers have been formulated as a general Integer Linear Programming (ILP) problem. [22], [9], [24] solved the ILP by relaxing the integral constraints to continuous constraints and optimizing a Linear Program, where the solution can be computed efficiently. A subsequent branch-and-cut method to find the global optimal to the ILP [24] or a simple rounding step [9] is used to acquire a final discrete solution. [34], [35] formulated tracking as clique partitioning, which can also be formulated as an ILP problem and solved by a heuristic clique merging method. [32] formulated tracking as a General Linear Assignment ILP problem, which was approximately solved with a deterministic annealing “softassign” algorithm [36].

More complex data association methods have also been used, including continuous energy minimization [13], discrete-continuous optimization [3], Block-ICM [2], conditional random fields [19], [14], generalized minimum clique [11] and quadratic programming [23], [37].

Even though each data association method has different merits, many of the aforementioned methods do not utilize actual person identity information such as face recognition, and in many cases it is non-trivial to incorporate the identity information into the previously proposed data association frameworks. One quick way to incorporate identity information may be to assign identities to trajectories after the trajectories have been computed. However, problems occur if two different identities are assigned to the same trajectory, and the true identity of the trajectory is no longer clear. Another approach may be to follow [7] and utilize the Viterbi algorithm to find a trajectory which passes through all the identity observations of each person. However, Viterbi search cannot be performed simultaneously over all individuals, and [7] proposed to perform Viterbi search sequentially, i.e. one individual after another. This greedy approach can lead to “hijacking” of another person’s trajectory [7], which is not ideal. Therefore, to achieve effective identity-aware tracking, it is more ideal to specially design a data association framework which can directly incorporate identity information into the optimization process.

Identity-Aware Data Association

Previously proposed data association methods [9], [38], [39] and [6] utilize identity information for tracking. There have been other work which utilizes transcripts from TV shows to perform face recognition and identity-aware face tracking [40], [41], but this is not the main focus of our paper.

[9], [38] formulated identity-aware tracking as an ILP and utilized person identification information from numbers written on an athlete’s jersey or from face recognition. Results show that the method is very effective in tracking basketball and soccer players, even when there are many occlusions. [9], [38] depends on the global appearance term for assigning identities to detections. However, the global term assumes a fixed appearance template for an object, which may not be applicable in surveillance scenes recorded over many hours as the appearance of the same person may change drastically.

[39] utilizes online structured learning to learn a target-specific model, which is used to compute the edge weights in a network flow framework. Though [39] has a stronger appearance model to compensate for drawbacks of network flow methods, it utilizes densely-sampled windows instead of person bounding boxes as input, which may be too time-consuming to compute in hundreds of hours long video sequences.

The work in [6] shows that a semi-supervised tracker which utilizes face-recognition as sparse label information for each class/individual achieves good tracking performance in a complex indoor environment. However, [6] does not incorporate the spatial locality constraint during the optimization step. Without the constraint, the solution acquired from the optimization step might show a person being in multiple places at the same time, thus this method does not work well for crowded scenes. Also, the method needs a Viterbi search to compute the final trajectories. The Viterbi search requires the start and end locations of all trajectories, which is an unrealistically restrictive assumption for long-term tracking scenarios. In this paper, we enhance this tracker by adding the spatial-locality constraint term, which enables tracking in crowded scenes and also removes the need for the start and end locations of a trajectory.

3 METHODOLOGY

Tracking-by-detection-based multi-object tracking can be viewed as a constrained clustering problem as shown in Figure 3. Each location hypothesis, which is a person detection result, can be viewed as a point in the spatial-temporal space, and our goal is to group the points so that the points in the same cluster belong to a single trajectory. A trajectory should follow the mutual exclusion constraint and spatial-locality constraint, which are defined as follows.

- **Mutual Exclusion Constraint:** a person detection result can only belong to at most one trajectory.
- **Spatial-Locality Constraint:** two person detection results belonging to a single trajectory should be reachable with reasonable velocity, i.e. a person cannot be in two places at the same time.

Sparse label information acquired from sources such as face recognition can be used to assign real-world identities and also enhance tracking performance.

To compute the trajectories, our tracking algorithm has three main steps.

- 1) **Manifold construction based on appearance and spatial affinity:** The appearance and spatial affinity respectively assumes that 1) similar looking person detections are likely to be of the same individual and 2) person detections which are spatially and temporally very close to each other are also likely to be of the same individual.
- 2) **Spatial locality constraint:** This constraint encodes the fact that a person cannot be at multiple places at the same time. In contrast to the manifold created in the previous step which encodes the *similarity* of two person detections, this constraint encodes the *dissimilarity* of two person detections.
- 3) **Constrained nonnegative optimization:** Our nonnegative optimization method acquires a solution which simultaneously satisfies the manifold assumption, the mutual exclusion constraint and the spatial-locality constraint.

In the following sections, we first define our notations, then the 3 aforementioned steps are detailed.

3.1 Notations

In this paper, given a matrix \mathbf{A} , let \mathbf{A}_{ij} denote the element on the i -th row and j -th column of \mathbf{A} . Let \mathbf{A}_i denote the i -th row of \mathbf{A} . $\text{Tr}(\cdot)$ denotes the trace operator. $\|\cdot\|_F$ is the Frobenius norm of a matrix. Given an arbitrary number m , $\mathbf{1}_m \in \mathbb{R}^m$ is a column vector with all ones.

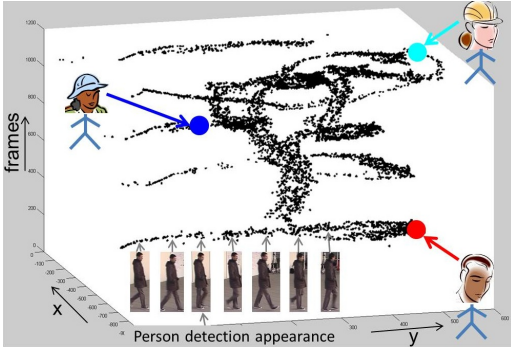
Hereafter, we call a person detection result a *data point*. Suppose the person detector detects n data points. Let c be the number of tracked individuals, which can be determined by either a pre-defined gallery of faces or the number of unique individuals identified by the face recognition algorithm. Our task is to assign a class label to each data point. Let $\mathbf{F} \in \mathbb{R}^{n \times c}$ be the label assignment matrix of all the data points. Without loss of generality, we assume that the data points are reorganized such that the data points from the same class are put together. The j -th column of \mathbf{F} is given by:

$$\mathbf{F}_{*j} = \left[\underbrace{0, \dots, 0}_{\sum_{i=1}^{j-1} m_{(i)}}, \underbrace{1, \dots, 1}_{m_{(j)}}, \underbrace{0, \dots, 0}_{\sum_{i=j+1}^c m_{(i)}} \right]^T, \quad (1)$$

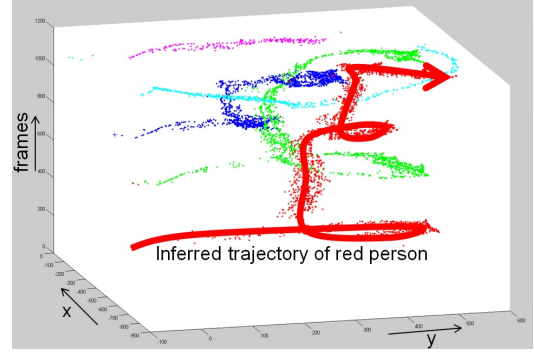
where $m_{(j)}$ is the number of data points in the j -th class. If the p -th element in \mathbf{F}_{*j} , i.e. \mathbf{F}_{pj} is 1, it indicates that the p -th data point corresponds to the j -th person. According to Equation 1, it can be verified that

$$\mathbf{F}^T \mathbf{F} = \begin{bmatrix} \mathbf{F}_{*1}^T \\ \vdots \\ \mathbf{F}_{*c}^T \end{bmatrix} [\mathbf{F}_{*1} \quad \dots \quad \mathbf{F}_{*c}] = \text{diag} \left(\begin{bmatrix} m_{(1)} \\ \vdots \\ m_{(c)} \end{bmatrix} \right) = \mathbf{J}, \quad (2)$$

where $\mathbf{J} \in \mathbb{R}^{c \times c}$. The i -th data point is described by a d dimensional color histogram $\mathbf{x}_{(i)} \in \mathbb{R}^d$, frame number $t_{(i)}$, and 3D location $\mathbf{p}_{(i)} \in \mathbb{R}^3$ which corresponds to the 3D location of the bottom center of the bounding box. In most cases, people walk on the ground plane, and the z component



(a) Input to tracking algorithm: location and appearance of person detection plus recognized faces for some person detections.



(b) Output of tracking algorithm: partitioning of the person detections into different trajectories.

Fig. 3: Illustration of the input and output of our tracking algorithm. Each person detection is a point in the (x, y, t) space. We assume that people walk on the ground plane, so the z axis is irrelevant. The figures are drawn based on the person detections from the *terrace1* data set [7].

becomes irrelevant. However, our method is not constrained to only tracking people on the ground plane.

3.2 Manifold Construction based on Appearance and Spatial Affinity

There are two aspects we would like to capture with manifold learning: 1) appearance affinity and 2) spatial affinity, which we will detail in the following sections.

3.2.1 Modeling Appearance Affinity

Appearance affinity assumes that if two data points are similar in appearance, then it is very likely that the two points correspond to the same person. This assumption can be captured with manifold learning, which is usually done in a two step process. First, we need to find suitable nearest neighbors for each data point. The assumption is that the nearest neighbors are highly likely to be of the same class as the current data point. Second, we take the nearest neighbor information of each point and encode the manifold structure into the Laplacian matrix.

Given a data point, suitable nearest neighbors are other similar-looking data points which are spatially and temporally “nearby”. More specifically, for the i -th data point, let the set of spatial-temporal neighbors be $\mathcal{M}_{(i)}$. $\mathcal{M}_{(i)}$ contains data points which are not only less than T frames away from the point, but also reachable from location $p_{(i)}$ with reasonable velocity. To avoid edge cases in computing velocity, we define velocity between data points i and l as follows:

$$v_{(il)} = \frac{\max(\|\mathbf{p}_{(i)} - \mathbf{p}_{(l)}\|_2 - \delta, 0)}{|t_{(i)} - t_{(l)}| + \epsilon}. \quad (3)$$

ϵ is a small number to avoid division by zero. δ models the maximum localization error of the same person from different cameras due to calibration and person detection errors, so when $t_{(i)} = t_{(l)}$, if the two data points are less than δ apart, these two points are still spatial-temporal neighbors. Therefore, $\mathcal{M}_{(i)}$ is defined as follows:

$$\mathcal{M}_{(i)} = \{l \mid v_{(il)} \leq V, |t_{(i)} - t_{(l)}| \leq T, 1 \leq l \leq n\}, \quad (4)$$

where V is the maximum possible velocity of a moving person. If the velocity required to move between two points is too large, then the two points cannot be of the same individual. Given $\mathcal{M}_{(i)}$, we look for data points in the set which have color histograms similar to data point i , as it is likely these points will belong to the same physical individual. To compute the similarity between two color histograms \mathbf{H}_i and \mathbf{H}_j , the exponential- χ^2 metric is used:

$$\chi^2(\mathbf{H}_i, \mathbf{H}_j) = \exp\left(-\frac{1}{2} \sum_{l=1}^d \frac{(\mathbf{H}_{il} - \mathbf{H}_{jl})^2}{\mathbf{H}_{il} + \mathbf{H}_{jl}}\right), \quad (5)$$

Based on Equation 5, two color histograms are similar only if their similarity is above a certain threshold γ . Finally, the set of nearest neighbors for data point i is found by selecting the top k nearest neighbors in $\mathcal{M}_{(i)}$ which have a similarity score larger than γ . We denote this set as $\mathcal{N}_{(i)} \subset \mathcal{M}_{(i)}$.

This method of finding neighbors makes our tracker more robust to occlusions. Occlusions may cause the tracking target to be partially or completely occluded. However, the tracking target usually reappears after a few frames. Therefore, instead of trying to explicitly model occlusions, we try to connect the observations of the tracking target before and after the occlusion. As shown in Figure 4, despite heavy occlusions in a time segment, the algorithm can still link the correct detections after the occlusion. The window size T affects the tracker’s ability to recover from occlusions. If T is too small, the method will have difficulty recovering from occlusions that last longer than T . However, a large T may increase chances of linking two different objects.

Once the nearest neighbors for each data point have been computed, the manifold structure can be encoded with a Laplacian matrix as follows. We first compute the affinity matrix \mathbf{W} , where $\mathbf{W}_{ij} = \chi^2(\mathbf{H}_i, \mathbf{H}_j)$ if $j \in \mathcal{N}_{(i)}$ and 0 otherwise. Then, the diagonal degree matrix \mathbf{D} of \mathbf{W} is computed, i.e. $\mathbf{D}_{ii} = \sum_{l=1}^n \mathbf{W}_{il}$. Finally, the Laplacian matrix \mathbf{L} which captures the manifold structure in the appearance space is $\mathbf{L} = \mathbf{D} - \mathbf{W}$.

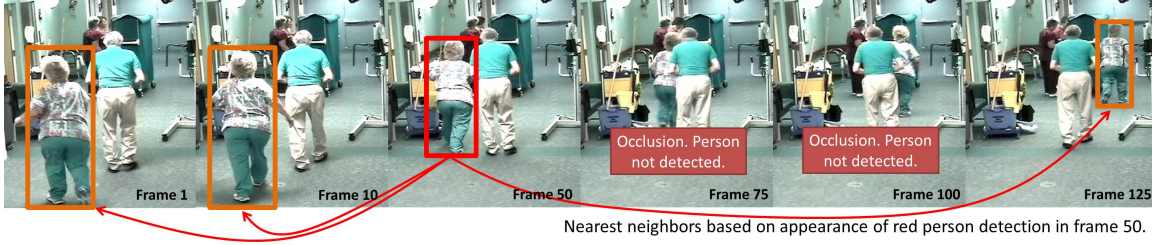


Fig. 4: Intuition of appearance-based nearest neighbor selection. The nearest neighbors for the red person detection in frame 50 are shown. No nearest neighbors are found in frames 75 and 100 as the person is occluded. Nevertheless, once the person is no longer occluded, the nearest neighbor connections can be made again, thus overcoming this occlusion.

3.2.2 Modeling Spatial Affinity

Other than modeling person detections of similar appearance, person detections which are “very close” (e.g. a few centimeters apart) in the same or neighboring frames are also very likely to belong to the same person. This assumption is reasonable in a multi-camera scenario because multiple detections will correspond to the same person, and due to calibration and person detection errors, not all detections will be projected to the exact same location. In this case, regardless of the appearance difference which may be resulting from non-color-calibrated cameras, these detections should belong to the same person. We therefore encode this information with another Laplacian matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ defined as follows. Let $\mathcal{K}_{(i)}$ be the set of data points which are less than distance \tilde{D} away and less than \tilde{T} frames away from point i , i.e.,

$$\mathcal{K}_{(i)} = \left\{ l \mid \|\mathbf{p}_{(i)} - \mathbf{p}_{(l)}\|_2 \leq \tilde{D}, |t_{(i)} - t_{(l)}| \leq \tilde{T}, 1 \leq l \leq n \right\}. \quad (6)$$

We compute the affinity matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ from $\mathcal{K}_{(i)}$ by setting $\mathbf{A}_{ij} = 1$ if $j \in \mathcal{K}_{(i)}$ and $\mathbf{A}_{ij} = 0$ otherwise. Define $\hat{\mathbf{D}} \in \mathbb{R}^{n \times n}$ as a diagonal matrix where $\hat{\mathbf{D}}_{ii}$ is the sum of \mathbf{A} ’s i -th row. Following [42], the normalized Laplacian matrix is computed: $\mathbf{K} = \mathbf{I} - \hat{\mathbf{D}}^{-\frac{1}{2}} \mathbf{A} \hat{\mathbf{D}}^{-\frac{1}{2}}$. The parameters \tilde{D} and \tilde{T} are all set very conservatively to avoid connecting to person detections from different individuals.

Then the loss function which combines the appearance and spatial affinity is as follows:

$$\begin{aligned} & \min_{\mathbf{F}} \text{Tr}(\mathbf{F}^T (\mathbf{L} + \mathbf{K}) \mathbf{F}) \\ & \text{s.t. columns of } \mathbf{F} \text{ satisfy Equation 1, } \forall i \in \mathcal{Y}, \mathbf{F}_i = \mathbf{Y}_i. \end{aligned} \quad (7)$$

Minimizing the loss term will result in a labeling which follows the manifold structure specified by appearance and spatial affinity. The first term in the constraint specifies that the label assignment matrix \mathbf{F} should be binary and have a single 1 per row. The second term in the constraints of the loss function is the face recognition constraint. Face recognition information is recorded in $\mathbf{Y} \in \mathbb{R}^{n \times c}$, where $\mathbf{Y}_{ij} = 1$ if the i -th data point belongs to class j , i.e. the face of data point i is recognized as person j . $\mathbf{Y}_{ij} = 0$ if we do not have any label information. There should only be at most a single 1 in each row of \mathbf{Y} . $\mathcal{Y} = \{i \mid \exists j \text{ s.t. } \mathbf{Y}_{ij} = 1\}$ are all the rows of \mathbf{Y} which have a non-zero element (i.e. a recognized face). As face recognition is approaching human-level performance [43], it is in most cases reasonable to treat it as a hard constraint.

Experiments analyzing the effect of face recognition errors on tracking performance are detailed in Section 4.1.7.

3.3 Spatial Locality Constraint

A person cannot be in multiple places at the same time. A tracker which cannot model this constraint, such as [6], might unreasonably state that a person is in multiple places at the same time. We incorporate the spatial locality constraint into our tracker by modeling pairwise person detection constraints. Given a pair of person detections (i, j) , if the speed $v_{(ij)}$ defined in Equation 3 required to move from one person detection to the other is too large, then it is highly unlikely that the pair of person detections will belong to the same person. We aggregate all the person detection pairs which are highly unlikely to be of the same individual and encode them in the matrix $\tilde{\mathbf{S}}$, as shown in Equation 8.

$$\tilde{\mathbf{S}}_{ij} = \begin{cases} 0 & \text{if } v_{(ij)} \leq V \\ 1 & \text{otherwise} \end{cases}, \quad 1 \leq i, j \leq n, \quad (8)$$

where V is the maximum possible velocity of a moving person. $\tilde{\mathbf{S}}$ is defined so that if none of the person detection velocity constraints were violated, then $\mathbf{F}_{*j}^T \tilde{\mathbf{S}} \mathbf{F}_{*j} = 0$, where \mathbf{F}_{*j} is the label assignment vector (column vector of \mathbf{F}) for the j -th person. We gather this constraint for all individuals and obtain $\text{Tr}(\mathbf{F}^T \tilde{\mathbf{S}} \mathbf{F}) = 0$ if none of the constraints were violated. The scale of $\tilde{\mathbf{S}}$ is normalized to facilitate the subsequent optimization step. Let \mathbf{D}' be a diagonal matrix where \mathbf{D}'_{ii} is the sum of row i of $\tilde{\mathbf{S}}$, then we can compute the normalized $\mathbf{S} = \mathbf{D}'^{-\frac{1}{2}} \tilde{\mathbf{S}} \mathbf{D}'^{-\frac{1}{2}}$. The spatial locality constraint is incorporated into our objective function as shown in Equation 9.

$$\begin{aligned} & \min_{\mathbf{F}} \text{Tr}(\mathbf{F}^T (\mathbf{L} + \mathbf{K}) \mathbf{F}) \quad \text{s.t. } \text{Tr}(\mathbf{F}^T \mathbf{S} \mathbf{F}) = 0, \\ & \text{columns of } \mathbf{F} \text{ satisfy Equation 1, } \forall i \in \mathcal{Y}, \mathbf{F}_i = \mathbf{Y}_i. \end{aligned} \quad (9)$$

Note that our spatial-locality constraint is a generalization to what is used in many single-camera multi-object network flow-based trackers [1], [12], where a person not being at two places at the same time is enforced by assuming that a trajectory can only be assigned to a single person detection at one time instant. However, in a multi-camera scenario, it is often the case that multiple detections from different cameras will correspond to the same individual, and the assumption used by network flow-based trackers may not be applicable here. Therefore, in this paper we propose a more general spatial-locality constraint, which can handle the case where multiple

detections from multiple cameras all correspond to the same individual. In the current formulation, we did not explicitly model the fact two detections from the same frame cannot belong to the same person, which could be easily added to our method. Also, experiments show that our method already achieves competitive results without this additional constraint, demonstrating the effectiveness of our spatial locality constraint.

Also note that the purpose of the affinity-based Laplacian matrix \mathbf{L} and \mathbf{K} are completely opposite of the purpose of \mathbf{S} . \mathbf{L} and \mathbf{K} specifies which two data points should be in the same cluster, while \mathbf{S} enforces the must-not-link constraint, i.e. these two points cannot be in the same cluster. Though both \mathbf{L} and \mathbf{S} utilize the assumption that a person cannot be at multiple places at the same time, these two matrices have completely different purpose in the loss function.

3.4 Nonnegative Matrix Optimization

Equation 9 is a combinatorial problem as the values of \mathbf{F} are limited to zeros and ones. This is very difficult to solve and certain relaxation is necessary to efficiently solve the objective function. Therefore, we first relax the form of Equation 9, and then an iterative projected nonnegative gradient descent procedure is utilized to optimize the relaxed loss function.

To perform relaxation, note that according to Equation 2, the columns of \mathbf{F} are orthogonal to each other, i.e. $\mathbf{F}^T \mathbf{F} = \mathbf{J}$ is a diagonal matrix. Also, \mathbf{F} is nonnegative by definition. According to [44], if both the orthogonal and nonnegative constraints are satisfied for a matrix, there will be at most one non-zero entry in each row of the matrix, which is still sufficient for discretizing \mathbf{F} and identifying the class-membership of each data point, i.e. the mutual exclusion constraint still holds. Therefore, we relax the form of \mathbf{F} , which originally is a binary label-assignment matrix, by only keeping the column orthogonal and nonnegative constraint. This leads to solving Equation 10.

$$\begin{aligned} \min_{\mathbf{F}} Tr(\mathbf{F}^T (\mathbf{L} + \mathbf{K}) \mathbf{F}) \\ s.t. Tr(\mathbf{F}^T \mathbf{S} \mathbf{F}) = 0, \mathbf{F}^T \mathbf{F} = \mathbf{J}, \mathbf{F} \geq 0, \forall i \in \mathcal{Y}, \mathbf{F}_i = \mathbf{Y}_i. \end{aligned} \quad (10)$$

Equation 10 is a constrained quadratic programming problem, in which the mutual exclusion constraint is enforced by $\mathbf{F}^T \mathbf{F} = \mathbf{J}$ and $\mathbf{F} \geq 0$. Under these constraints, the values in \mathbf{F} are continuous and no longer binary, but there will still only be *at most* one non-zero entry per row. One big advantage of this relaxation is that now our method can naturally handle false positive detections, because \mathbf{F} is now also allowed to have a row where all elements are zeros, which corresponds to a person detection not being assigned to any class.

$\mathbf{F}^T \mathbf{F} = \mathbf{J}$ is still a difficult constraint to optimize. If \mathbf{J} is the identity matrix, then $\mathbf{F}^T \mathbf{F} = \mathbf{I}$ forms the Stiefel manifold [45]. Though a few different methods have been proposed to perform optimization with the orthogonal constraint [45], [46], [47], [48], many methods require a specific form of the objective function for the optimization process to converge. Therefore, we instead employ the simple yet effective quadratic penalty method [44], [49] to optimize the

loss function. The quadratic penalty method incorporates the equality constraints into the loss function by adding a quadratic constraint violation error for each equality constraint. The amount of violation is scaled by a weight τ , which gradually increases as more iterations of the optimization are performed, thus forcing the optimization process to satisfy the constraints. More details on the convergence properties of the quadratic penalty method can be found in [49]. To solve Equation 10, we move the constraints $\mathbf{F}^T \mathbf{F} = \mathbf{J}$ and $Tr(\mathbf{F}^T \mathbf{S} \mathbf{F}) = 0$ into the loss function as a penalty term. We rewrite the objective function as follows:

$$\begin{aligned} \min_{\mathbf{F}} f(\mathbf{F}) = \min_{\mathbf{F}} Tr(\mathbf{F}^T (\mathbf{L} + \mathbf{K} + \tau \mathbf{S}) \mathbf{F}) + \tau \|\mathbf{F}^T \mathbf{F} - \mathbf{J}\|_F^2 \\ s.t. \mathbf{F} \geq 0, \forall i \in \mathcal{Y}, \mathbf{F}_i = \mathbf{Y}_i. \end{aligned} \quad (11)$$

For each τ , we minimize Equation 11 until convergence. Once converged, τ is multiplied by 2 and Equation 11 is minimized again.

To solve for Equation 11 given a fixed τ , we perform projected nonnegative gradient descent [50], which iteratively updates the solution at iteration l ($\mathbf{F}^{(l)}$) to $\mathbf{F}^{(l+1)}$ as follows:

$$\mathbf{F}^{(l+1)} = P \left[\mathbf{F}^{(l)} - \alpha^{(l)} \nabla f(\mathbf{F}^{(l)}) \right] \quad (12)$$

where the projection function P :

$$P[\mathbf{F}_{ij}] = \begin{cases} \mathbf{F}_{ij} & \text{if } \mathbf{F}_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}, \quad (13)$$

is an element-wise function which maps an element back to the feasible region, i.e. in this case a negative number to zero. The step size $\alpha^{(l)}$ is found in a line search-like fashion, where we search for an $\alpha^{(l)}$ which provides sufficient decrease in the function value:

$$f(\mathbf{F}^{(l+1)}) - f(\mathbf{F}^{(l)}) \leq \sigma Tr \left(\nabla f(\mathbf{F}^{(l)})^T (\mathbf{F}^{(l+1)} - \mathbf{F}^{(l)}) \right). \quad (14)$$

Following [50], $\sigma = 0.01$ in our experiments. The gradient of our loss function f is

$$\nabla f(\mathbf{F}) = 2(\mathbf{L} + \mathbf{K} + \tau \mathbf{S}) \mathbf{F} + 4\tau \mathbf{F} (\mathbf{F}^T \mathbf{F} - \mathbf{J}). \quad (15)$$

Details on convergence guarantees are shown in [50]. To satisfy the face recognition constraints, the values of \mathbf{F} for the rows in \mathcal{Y} are set according to \mathbf{Y} and never updated by the gradient.

The main advantage of projected nonnegative gradient descent over the popular multiplicative updates for nonnegative matrix factorization [51], [45] is that elements with zero values will have the opportunity to be non-zero in later iterations. However, for multiplicative updates, zero values will always stay zero. In our scenario, this means that if $\mathbf{F}_{ij}^{(l)}$ shrinks to 0 at iteration l in the optimization process, the decision that “data point i is not individual j ” is final and cannot be changed, which is not ideal. The projected nonnegative gradient descent method does not have this issue as the updates are additive and not multiplicative.

\mathbf{J} is a diagonal matrix, where each element on the diagonal \mathbf{J}_{ii} corresponds to the number of data points belonging to class i , i.e. m_i . As m_i is unknown beforehand, m_i is estimated by the number of recognized faces belonging to class i plus a

Data: Location hypothesis $\mathbf{p}_{(i)}$, $t_{(i)}$, and appearance $\mathbf{x}_{(i)}$, $1 \leq i \leq n$. Face recognition matrix $\mathbf{Y} \in \mathbb{R}^{n \times c}$.

Result: Final label assignment matrix \mathbf{F}

```

Compute Laplacian matrices  $\mathbf{L}$ ,  $\mathbf{K}$  ;           // Sec. 3.2
Compute spatial locality matrix  $\mathbf{S}$  ;           // Sec. 3.3
Compute diagonal matrix  $\mathbf{J}$  ;                 // Sec. 3.4
Compute diagonal matrix  $\mathbf{U}$  from  $\mathbf{Y}$  ;         // Sec. 3.4
Initialize  $\mathbf{F}^{(0)}$  with Equation 16 ;
 $l \leftarrow 0$  ;                               // iteration count
 $\tau \leftarrow 10^{-4}$  ;                         // initial penalty
repeat // Solve for Equation 11 with penalty method
     $\tau \leftarrow \tau \times 2$  ;                   // gradually increase penalty  $\tau$ 
    repeat // projected gradient descent
        Compute  $\mathbf{F}^{(l+1)}$  from  $\mathbf{F}^{(l)}$  with Equation 12;
         $l \leftarrow l + 1$ ;
    until convergence;
until  $\tau \geq 10^{11}$ ;
return  $\mathbf{F}^{(l)}$ 

```

Algorithm 1: Main steps in proposed tracking algorithm.

constant β , which is proportional to the number of data points n . In our experiments we set $\beta = \frac{n}{1000}$.

To initialize our method, we temporarily ignore the mutual exclusion and spatial locality constraint and only use the manifold and face recognition information to find the initial value $\mathbf{F}^{(0)}$. $\mathbf{F}^{(0)}$ is obtained by minimizing Equation 16.

$$\min_{\mathbf{F}^{(0)}} Tr \left((\mathbf{F}^{(0)})^T (\mathbf{L} + \mathbf{K}) \mathbf{F}^{(0)} + (\mathbf{F}^{(0)} - \mathbf{Y})^T \mathbf{U} (\mathbf{F}^{(0)} - \mathbf{Y}) \right). \quad (16)$$

$\mathbf{U} \in \mathbb{R}^{n \times n}$ is a diagonal matrix. $\mathbf{U}_{ii} = \infty$ (a large constant) if $i \in \mathcal{Y}$, i.e. the i -th data point is a ground truth positive for any class. Otherwise $\mathbf{U}_{ii} = 1$. \mathbf{U} is used to enforce the consistency between prediction results and face recognition label information. The global optimal solution for Equation 16 is $\mathbf{F}^{(0)} = (\mathbf{L} + \mathbf{K} + \mathbf{U})^{-1} \mathbf{U} \mathbf{Y}$ [52].

Finally, once the optimization is complete, we acquire a \mathbf{F} which satisfies the mutual exclusion and spatial locality constraint. Therefore, trajectories can be computed by simply connecting neighboring observations belonging to the same class. At one time instant, if there are multiple detections assigned to a person, which is common in multi-camera scenarios, then the weighted average location is computed. The weights are based on the scores in the final solution of \mathbf{F} . A simple filtering process is also utilized to remove sporadic predictions. The main steps of our tracker are summarized in Algorithm 1.

4 EXPERIMENTS

We present experiments on short-term and long-term tracking followed by video summarization experiments based on our long-term tracking output.

4.1 Short-term and Long-term Tracking

4.1.1 Data Sets

As we are interested in evaluating identity-aware tracking, we focused on sequences where identity information such

as face recognition was available. Therefore, many popular tracking sequences such as the PETS 2009 sequences [53], Virat [54], TRECVID 2008 [55] and Town Centre [56] were not applicable as the faces in these sequences were too small to be recognized. The following three data sets were utilized in our experiments.

terrace1: The 4 camera *terrace1* [7] data set has 9 people walking around in a 7.5m by 11m area for 5000 frames under 25fps, which corresponds to a total of around 13 minutes of video. The scene is very crowded, thus putting the spatial locality constraint to test. The POM grid we computed had width and height of 25 centimeters per cell. Person detections were extracted for every frame. As the resolution of the video is low, one person did not have a recognizable face. For the sake of performing identity-aware tracking on this dataset, we manually added two identity annotations for each individual at the start and end of the person's trajectory to guarantee that each individual had identity labels. None of the trackers utilized the fact that these two additional annotations were the start and end of a trajectory. In total, there were 794 identity labels out of 57,202 person detections.

Caremedia Short: The 15 camera *Caremedia Short* [57], [6] data set has 13 individuals performing daily activities in a nursing home for 11310 frames under 30 fps, which corresponds to a total of around 94 minutes of video. The data set was first used in [6]. Manual annotations were provided every second and further interpolated to every frame. The 15 surveillance cameras were set up on the ceilings of public areas in a nursing home. There are many occlusions caused by walls, which is typical of indoor scenes. There are also many challenging scenes such as long corridors with sparse camera setups, where considerable occlusion was observed. In addition, there is no single camera which has a global view of the whole environment, which is typical of many surveillance camera setups, but atypical of the data sets that have been used to perform multi-camera tracking. The data set records activities in a nursing home where staff maintain the nursing home and assist residents throughout the day. As the data set covers a larger area and is also longer than *terrace1*, we ran into memory issues for trackers which take POM as input when our cell size was 25 centimeters. Therefore, the POM grid we computed had width and height of 40 centimeters per cell. Person detections were extracted from every sixth frame. In total, there were 2,808 recognized faces out of 12,129 person detections. Though on average there was a face for every 4 person detections, but recognized faces are usually found in clusters and not evenly spread out over time. So there were still long periods of time when no faces were recognized.

Caremedia Long: The 15 camera *Caremedia Long* data set is a newly annotated data set which has 49 individuals performing daily activities in the same nursing home as *Caremedia Short*. There are 116.25 hours of video in total, which corresponds to 7 hours 45 minutes of wall time. To the best of our knowledge, this is one of the longest sequence to date to be utilized for multi-object tracking experiments, thus enabling us to evaluate tracking algorithms in realistic long-term tracking scenarios. Ground truth was annotated every minute. Person detections were extracted from every sixth frame. In total, there were

70,994 recognized faces out of 402,833 person detections.

4.1.2 Baselines

We compared our method with three identity-aware tracking baselines. As discussed in the Related Work section (Section 2), it is non-trivial to modify a non-identity-aware tracker to incorporate identity information. Therefore, other trackers which did not have the ability to incorporate identity information were not compared.

Multi-Commodity Network Flow (MCNF): The MCNF tracker [9] can be viewed as an extension of the K-Shortest-Path tracker (KSP, [8]) with identity aware capabilities. The KSP is a network flow-based method that utilizes localization information based on POM. Given the POM localizations, a network flow graph is formed. The algorithm will then find the K shortest paths to the graph, which correspond to the K most likely trajectories in the scene. MCNF further duplicates the graph in KSP for every different *identity group* in the scene. The problem is solved with linear programming plus an additional step of rounding non-integral values.

Following [9], we reimplemented the MCNF tracker. In our experiments, the graph is duplicated c times, because for our setup each individual belongs to its own identity group. Gurobi [58] was used as our linear program solver. Global appearance templates of each person were computed from the appearance of person detections which had recognized faces. Occlusions were computed from a raw probability occupancy map, and occluded observations were not used to generate templates nor compare color histograms. Following [9], the input of MCNF was taken from the output of POM and KSP to save computation time. The source code of POM and KSP were from the authors [7], [8]. For trajectories which came closer to three grid cells, the cells in between the two trajectories were also activated so that the MCNF had the freedom to switch trajectories if necessary. This setting is referred to as *MCNF w/ POM*. The base cost of generating a trajectory, which is a parameter that controls the minimum length of the generated tracks, is set to -185 for all *MCNF w/ POM* experiments.

For the two Caremedia data sets, we also took the person detection (PD) output and generated POM-like localization results which were also provided to MCNF. The POM-like localization results were generated by first creating a grid for the nursing home scene, and then aggregating all person detections falling into each grid at each time instant. This setting is referred to as *MCNF w/ PD*. For all *MCNF w/ PD* experiments, the grid size is 40 centimeters, the base cost of generating a trajectory is -60, and detections were aggregated over a time span of 6 frames to prevent broken trajectories. For the *Caremedia Long* set, the Gurobi solver was run in 12,000 frame batches to avoid memory issues.

Lagrangian Relaxation (LR): [39] utilizes LR to impose mutual exclusion constraints for identity-aware tracking in a network flow framework very similar to MCNF, where each identity has their own identity specific edges. Lagrange multipliers enforce the mutual exclusion constraint over mutual-exclusive edges in the graph. To optimize with LR, dynamic programming first finds trajectories for each identity given the current network weights. Then, the Lagrange multipliers,

which are a part of the network weights, are updated to penalize observations that violate the mutual exclusion constraint. This process is repeated again on the updated network weights till convergence. To fairly compare different data association methods, our LR-based tracker utilizes the same appearance information used by all our other trackers, thus the structured learning and densely sampled windows proposed in [39] were not used. Specifically, LR uses the same POM-like input and network as MCNF.

Non-Negative Discretization (NND): The Non-Negative Discretization tracker [6] is a primitive version of our proposed tracker. The two main differences are: 1) NND does not have the spatial locality constraint, thus an extra Viterbi trajectory formulation step is necessary, which requires the start and end of trajectories, and 2) a multiplicative update was used to perform non-negative matrix factorization. NND requires the start and end locations of trajectories, which are usually not available in real world scenarios. In our experiments, therefore, no start and end locations were provided to NND, and the final trajectories of NND were formed with the same method used by our proposed tracker. NND utilizes [52] to build the manifold, but internal experiments have shown that utilizing the method in [52] to build the Laplacian matrix achieves similar tracking performance compared to the standard method [42], [59]. Therefore, to fairly compare the two data association methods, we utilized the same Laplacian matrix computation method for NND and our method. Also the spatial affinity term \mathbf{K} was not used in the originally proposed NND, but for fairness we add the \mathbf{K} term to NND.

4.1.3 Implementation Details

We utilized the person detection model from [60], [61] for person detection. The person detection results from different camera views were mapped to a common 3D coordinate system using the camera calibration and ground plane parameters provided. Color histograms for the person detection were computed the same way as in [6]. We used HSV color histograms as done in [4]. We split the bounding box horizontally into regions and computed the color histogram for each region similar to the spatial pyramid matching technique [62]. Given L layers, we have $2^L - 1$ partitions for each template. L was 3 in our experiments. Since the person detector only detects upright people, tracking was not performed on sitting people or residents in wheelchairs. Background subtraction for POM was performed with [63].

Face information is acquired from the PittPatt software¹, which can recognize a face when a person is close enough to the camera. We assumed that the gallery for the people we are interested in tracking is provided. There are two options to collect this gallery: 1) manually collect faces of the person or 2) perform face clustering over all detected faces and select clusters which consist of faces corresponding to the person-of-interest. Though the selection requires some manual effort, it does not consume a lot of time as the majority of faces have already been clustered. The latter option was utilized to create the *Caremedia* tracking sequences.

1. Pittsburgh Pattern Recognition (<http://www.pittpatt.com>)

For our proposed method, the parameters for all three data sets were as follows. The number of nearest neighbors used for appearance-based manifold construction was $k = 25$. The window to search for appearance-based nearest neighbors was $T = 8$ seconds. The color histogram threshold $\gamma = 0.85$. The maximum localization error $\delta = 125$ to take into account camera calibration errors. For modeling spatial affinity, \tilde{D} was 20 centimeters, and \tilde{T} was 6 frames. When computing the spatial locality constraint matrix \mathbf{S} , we found that computing the velocity between all pairs of data points will make the \mathbf{S} matrix very dense, thus we only looked for conflicting pairs of data points which were less than 6 frames apart to retain sparse \mathbf{S} . The above parameters were also used for NND. For the optimization step, the initial value of $\tau = 2 \times 10^{-4}$, and the final value was $\tau = 10^{11}$.

4.1.4 Evaluation Metrics

Identity-aware tracking can be evaluated from a multi-object tracking point of view and a classification point of view. From the tracking point of view, the most commonly used multi-object tracking metric is *Multiple Object Tracking Accuracy* (MOTA²) [64], [65]. Following the evaluation method used in [3], [6], the association between the tracking results and the ground truth is computed in 3D with a hit/miss threshold of 1 meter. MOTA takes into account the number of true positives (TP), false positives (FP), missed detections (false negatives, FN) and identity switches (ID-S). Following the setting in [9] ³ MOTA is computed as follows:

$$\text{MOTA} = 1 - \frac{\# \text{FP} + \# \text{FN} + \log_{10}(\# \text{ID-S})}{\# \text{ground truth}}. \quad (17)$$

However, the TP count in MOTA does not take into account the identity of a person, which is unreasonable for identity aware tracking. Therefore, we compute identity-aware true positives (I-TP), which means that a detection is only a true positive if 1) it is less than 1 meter from the ground-truth and 2) the identities match. Similarly, we can compute I-FP and I-MD, which enables us to compute classification-based metrics such as micro-precision ($\text{MP} = \frac{\# \text{I-TP}}{\# \text{I-TP} + \# \text{I-FP}}$), micro-recall ($\text{MR} = \frac{\# \text{I-TP}}{\# \text{I-TP} + \# \text{I-FN}}$) and a comprehensive micro-F1 ($\frac{2 \times \text{MP} \times \text{MR}}{\text{MP} + \text{MR}}$) for each tracker. The *micro*-based performance evaluation takes into account the length (in terms of time) of each person’s trajectory, so a person who appears more often has larger influence to the final scores.

4.1.5 Tracking Results

Tracking results for the three data sets are shown in Table 1. As we are more interested in identity-aware tracking, we pay more attention to the comprehensive F1-score from the classification-based metrics, which will only be high if both precision and recall are high. We achieve the best performance in F1-scores across all three data sets. This means that our tracker can not only track a person well, but can also accurately

Method	Micro-Precision	Micro-Recall	Micro-F1	TP	FN	FP	ID-S	MOTA
KSP w/ POM	N/A	N/A	N/A	22182	2990	767	187	0.852
MCNF w/ POM	0.593	0.532	0.561	21864	3298	644	197	0.844
LR w/ POM	0.609	0.478	0.535	19216	5996	521	147	0.743
NND	0.613	0.238	0.343	8035	17267	1771	57	0.249
Ours w/o SLC	0.704	0.346	0.464	10642	14655	1745	62	0.353
Ours	0.692	0.635	0.663	21370	3873	1783	116	0.777

(a) Tracking performance on *terrace1* sequence.

Method	Micro-Precision	Micro-Recall	Micro-F1	TP	FN	FP	ID-S	MOTA
KSP w/ POM	N/A	N/A	N/A	21286	11794	36035	939	-0.406
MCNF w/ POM	0.117	0.238	0.157	23493	9769	44452	757	-0.594
MCNF w/ PD	0.746	0.578	0.652	19941	13749	5927	329	0.422
LR w/ PD	0.802	0.565	0.663	19415	14408	4203	196	0.453
NND	0.861	0.726	0.787	25628	8364	3100	27	0.663
Ours w/o SLC	0.869	0.726	0.791	25578	8408	3080	33	0.662
Ours	0.865	0.755	0.807	26384	7576	3537	59	0.673

(b) Tracking performance on *Caremedia Short* sequence.

Method	Micro-Precision	Micro-Recall	Micro-F1	TP	FN	FP	ID-S	MOTA
MCNF w/ PD	0.743	0.418	0.535	265	347	71	25	0.342
LR w/ PD	0.787	0.405	0.535	261	360	52	16	0.351
NND	0.588	0.505	0.543	314	281	174	42	0.283
Ours w/o SLC	0.638	0.549	0.590	349	257	151	31	0.357
Ours	0.648	0.571	0.607	370	241	149	26	0.386

(c) Tracking performance on *Caremedia Long* sequence.

TABLE 1: Tracking performance on 3 tracking sequences. POM: Probabilistic Occupancy Map proposed in [7] as input. PD: Person detection as input. SLC: Spatial locality constraint. “w/” and “w/o” are shorthand for “with” and “without” respectively. We did not perform the *MCNF w/ POM* on the *Caremedia Long* sequence as it was already performing poorly on the shorter sequence.

identify the individual. Figure 5 and Figure 6 show qualitative examples of our tracking result.

The importance of the spatial locality constraint (SLC) is also shown clearly in Table 1a. Without the spatial locality constraint in the optimization step (*NND* and *Ours w/o SLC*), performance degraded significantly in the very crowded *terrace1* sequence as the final result may show a person being at multiple places at the same time, thus hijacking the person detections of other individuals. For the *Caremedia* sequences, the SLC does not make a big difference, because 1) the scene is not so crowded and 2) the appearance of each individual is more distinct, thus relying only on the appearance feature can already achieve good performance.

The MCNF tracker is also a very strong baseline. For *terrace1*, KSP and consequently MCNF achieved very good MOTA results with POM person localization. MCNF was slightly worse than KSP on MOTA scores because though MCNF is initialized by KSP, MCNF is no longer solving a problem with a global optimal solution. However, for the *Caremedia Short* sequence, results were poor. Through manual analysis, we found that POM, which is used for person localization in KSP and MCNF, has more difficulty in localizing on long corridors where cameras only view the principal direction of the corridor. For example, when there are multiple people on the corridor, their foreground masks tend to merge together into a single blob. Thus there will be many different ways the generative POM model can synthesize the foreground image for each camera view. This leads to ambiguities in person localization, which will significantly hurt tracking

2. Code modified from <http://www.micc.unifi.it/lisanti/source-code/>.

3. There are two common transformation functions (denoted as $c_s()$ in [65]) for the identity-switch term, either \log_{10} [65], [9] or the identity function [64]. We have selected the former as this is what was used in MCNF, which is one of our baselines.



Fig. 5: Snapshots of tracking results from the 4 camera *terrace1* sequence.

performance. Cameras with a side-view on the corridor will significantly alleviate this issue, but this is usually not available in a corridor setting. Also, the indoor scene of *Caremedia Short* is more complex than *terrace1*. Therefore, even though there are 15 cameras in *Caremedia Short*, occlusions caused by walls mean that the camera coverage is not as perfect as *terrace1*, thus causing more ambiguities in POM localization. Lastly, there were other non-person moving objects such as carts and rolling closets in the scene, which would also be detected by POM. These ambiguities cause false positives, leading to poor KSP performance. As MCNF input is based on KSP output, *MCNF w/ POM* also performed poorly. However, this is unfair, as MCNF performed poorly due to inaccurate localization, which is unrelated to its data association method. Therefore, to fairly compare MCNF, we provided MCNF with the same localization information used by our method and ran the *MCNF w/ PD* experiment. With the new localization based on person detections, MCNF was able to achieve competitive results. This experiment shows that even if a tracker performs poorly, it may simply be due to a single malfunctioning component in the tracker, and if the component is switched with another effective component, the tracker can still achieve good results. We believe this is a fairer way of comparing different trackers. Nevertheless, with the same person detection as input, our method still outperforms MCNF in all sequences in terms of F1-score.

For long-term experiments, our best tracker can locate a person 57.1% of the time with 64.8% precision over 7 hours 45 minutes wall time of video. These results are encouraging, as the tracking output with such performance already has the potential to be utilized by other tasks, such as the experiments performed in Section 4.2 for surveillance video summarization.

4.1.6 Discussion - Advantages of Tracker

The key advantages of our tracker are as follows:

Face recognition information is integrated into the framework: Face recognition serves as a natural way to automatically assign identities to trajectories in long-term tracking scenarios, where manual intervention is prohibitively costly. When the tracker loses track of a person, face recognition

can also aid in automatic reinitialization. Also, in long-term scenarios, it is common that people will change clothes, thus drastically changing their appearance. Face recognition will still be able to recognize this person, making our method robust to large appearance changes of a single person.

Naturally handle appearance changes: Handling color appearance changes is crucial because the appearance of the tracking target can change gradually in different parts of the scene. In our tracker, the appearance templates of the tracked target are implicitly encoded in the manifold structure we learn. Therefore, if the appearance of a tracked object changes smoothly along a manifold, our algorithm can model the change. No fixed global appearance model is required to track each individual, and no threshold is required to decide when to adaptively update the appearance model. If there is a drastic change in appearance for a tracked object, then the appearance manifold will highly likely be broken as the nearest neighbor search will not select the detection where the object's appearance changed drastically. However, the spatial affinity Laplacian matrix \mathbf{K} still can potentially link up these two observations.

Take into account appearance from multiple neighbors: Our tracker models appearance by taking into account the appearance information from multiple neighboring points, which enables us to have a more stable model of appearance. Linear programming and network flow-based methods can only either model appearance globally and assume the appearance of a target will not change, or model appearance similarity only over the previous and next detection in the track.

Handle multiple detections per frame for one individual: In multi-camera scenes, it is common that at one time instant, multiple detections from different cameras correspond to the same physical person. This phenomenon may be difficult to deal with for single-camera multi-object trackers based on network flow [1], [12], because the spatial locality constraint for these methods are enforced based on the assumption that each individual can only be assigned a single person detection per frame. Therefore, multi-camera network flow-based methods such as [8], [9] utilize a two step process where the POM is first used to aggregate evidences from multiple cameras to perform localization. Then the data association step is used to compute trajectories. The two steps are necessary so that the spatial locality constraint can be enforced for network flow methods. In our case, our formulation of the spatial locality constraint, which is based on the velocity to travel between two detections being under a certain threshold, can be viewed as a generalization to the aforementioned assumption, and this enables us to combine the localization and data association steps in a single optimization framework.

No discretization of the space required in multi-camera scenarios: Previous multi-camera network flow methods [8], [9] requires discretization of the tracking space in multi-camera scenarios to make the computation feasible. Finer grids run into memory issues when the tracking sequence is long and covers a wide area, and coarser grids run the risk of losing precision. However, our tracker works directly on person detections, and discretization is not necessary.

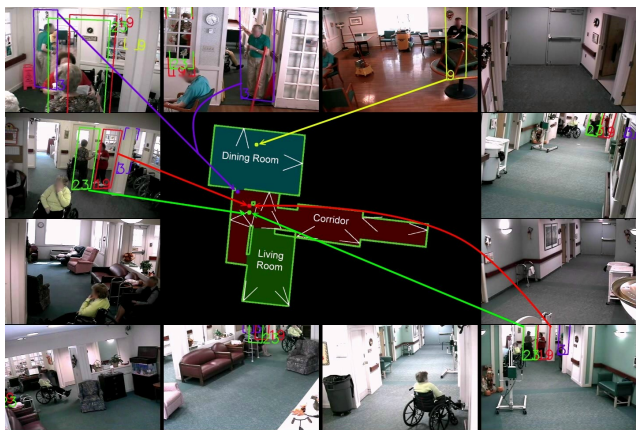


Fig. 6: Snapshots of tracking results from *Caremedia Long* data set. To increase readability, not all arrows are drawn and only 12 out of 15 cameras are shown.

4.1.7 Discussion - Limitations of Tracker

There are also limitations to our tracker.

Assumes at least one face recognition per trajectory: If there is a trajectory where no faces are observed and recognized, then our tracker will completely ignore this trajectory, which is acceptable if we are only interested in identity-aware tracking. Otherwise, one potential solution is to find clusters of unassigned person detections and assign pseudo-identities to them to recover the trajectories.

Only bounded velocity model employed: To employ the more sophisticated constant velocity model, we could use pairs of points as the unit of location hypotheses, but this may generate significantly more location hypotheses than the current approach.

Assumes all cameras are calibrated: To perform multi-camera tracking, we first map all person detections into a global coordinate system. In order to do so, the intrinsic and extrinsic parameters of the cameras need to be provided. If a camera moves, the updated extrinsic parameters also needs to be provided.

Face recognition gallery required beforehand: In order to track persons-of-interest, we require the gallery beforehand. This is the only manual step in our whole system, which could be alleviated when the detected faces are clustered thus making it very efficient for humans to map the face clusters to persons-of-interest. Also, in a nursing home setting, the people we are interested in tracking and observing are fixed, thus this is a one time effort which could be used for days, weeks or even months of recordings.

Assumes perfect face recognition: The current framework assumes perfect face recognition, which may not be applicable in all scenarios. Therefore, we also analyzed the effect of face recognition accuracy on tracking performance. Experiments were performed by mimicking face recognition errors through randomly corrupting the face recognition results in the *Caremedia Short* set. The original *Caremedia Short* set had around 2% face recognition error rate. In other scenarios, we may not be able to achieve such high face recognition accuracy, thus based on the *Caremedia Short* face recognitions, we generated face recognitions with error rate ranging from

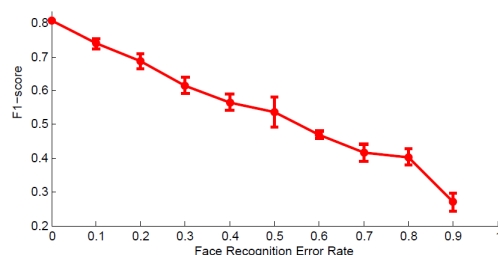
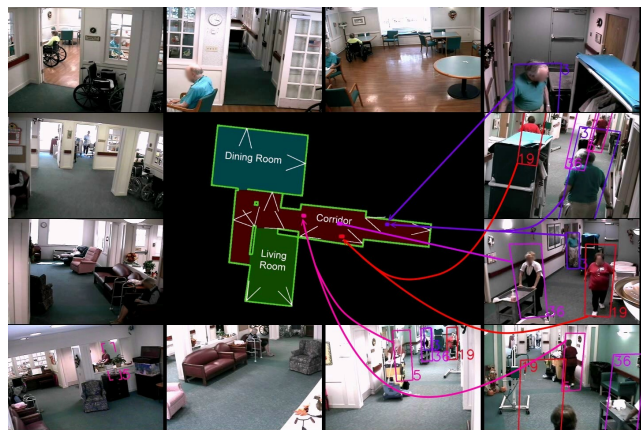


Fig. 7: Performance of tracking performance under varying face recognition error rate.

10% to 90%. The experiment was repeated 3 times per error rate, and the results with the 95% confidence intervals are shown in Figure 7. Results show that the general trend is a 20% increase in face recognition error rate will cause around 10% drop in tracking F1-score.

4.1.8 Timing Analysis

We analyzed the timing of each step in the tracking pipeline, which includes person detection, face recognition, color histogram extraction and long-term tracking. Currently, the most time consuming part is person detection. It took around 8 seconds per frame to run person detection from [60], [61] on a single core of an Intel Xeon E5649 2.53GHz CPU. We ran person detection on every sixth frame for 116.25 hours (*Caremedia Long*) of 30fps video, taking around 4650 core hours. However, person detection speed can be significantly sped up by recently proposed real-time person detectors [66], which will enable us to reduce the computation to 116.25 hours. The rest of the pipeline, which includes face recognition, color histogram extraction, and long-term tracking, took less than 300 core hours to process *Caremedia Long*. Specifically, given the person detection, face recognition information and color histograms, it took 2.7 hours to run our tracker on 116.25 hours of surveillance video with a 6 core Intel Xeon E5649 2.53GHz CPU. A total of 1.7 hours were spent on computing the Laplacian matrices, and 1 hour was spent on the optimization phase. In sum, with all the input features computed, the tracker runs at around $\frac{1}{40}$ times real-time, which is very fast.

4.2 Visual Diary Generation

To demonstrate the usefulness of our tracking output, video summarization experiments were performed. We propose to summarize surveillance video using visual diaries, specifically in the context of monitoring elderly residents in a nursing home. Visual diary generation for elderly nursing home residents enables doctors and staff to quickly understand the activities of a senior person throughout the day to facilitate the diagnosis of the elderly person’s state of health. The visual diary for a specific person consists of two parts as shown in Figure 2: 1) snippets which contain snapshots and textual descriptions of activities-of-interest performed by the person, and 2) activity-related statistics accumulated over the whole day. The textual descriptions of the detected events enables efficient indexing of what a person did at different times. The statistics for the activities detected can be accumulated over many days to discover long-term patterns.

We propose to generate visual diaries with a summarization-by-tracking framework. Using the trajectories acquired from our tracking algorithm, we extract motion patterns from the trajectories to detect certain activities performed by each person in the scene. The motion patterns are defined in a simple rule-based manner. Even though more complex methods such as variants of Hidden Markov Models [67] to detect interactions could also be used, our goal here is to demonstrate the usefulness of our tracking result and not test state-of-the-art interaction detection methods, thus only a simple method was used. The activities we detect are as follows:

- Room change: Given the tracking output, we can detect when someone enters or leaves a room.
- Sit down / stand up: We trained a sitting detector [61] which detects whether someone is sitting. Our algorithm looks for tracks which end/begin near a seat and check whether someone sat down/stood up around the same time.
- Static interaction: If two people stand closer than distance D' for duration T' , then it is likely that they are interacting.
- Dynamic interaction: If two people are moving with distance less than D' apart for a duration longer than T' , and if they are moving faster than 20 cm/s, then it is highly likely that they are walking together.

According to [68], if people are travelling in a group, then they should be at most 7 feet apart. Therefore, we set the maximum distance D' for there to be interaction between two people at 7 feet. The minimum duration of interaction T' was set to 8 seconds in our experiments.

Given the time and location of all the detected activities, we can sort the activities according to time and generate the visual diary. The visual diary for a given individual consists of the following:

- Snippets: snapshots and textual descriptions of the activity. Snapshots are extracted from video frames during the interaction and textual descriptions are generated using natural language templates.
- Room/state timing estimates: time spent sitting or standing/walking in each room.

Visual diary components	Micro-Precision	Micro-Recall	Micro-F1
Snippet generation	0.382	0.522	0.441
Room/state timing estimates	0.809	0.511	0.626
Interaction timing estimates	0.285	0.341	0.311
Interacting target prediction	0.533	0.762	0.627

TABLE 2: Evaluation of generated visual diary.

- Total interaction time: time spent in social interactions.
- Interacting targets: people with whom the person interacted.

Our proposed method of using tracking output for activity detection can be easily combined with traditional activity recognition techniques using low-level features such as Improved Dense Trajectories [69] with Fisher Vectors [70] to achieve better activity detection performance and detect more complex actions, but extending activity recognition to activity detection is beyond the scope of this paper.

Visual Diary Generation Results

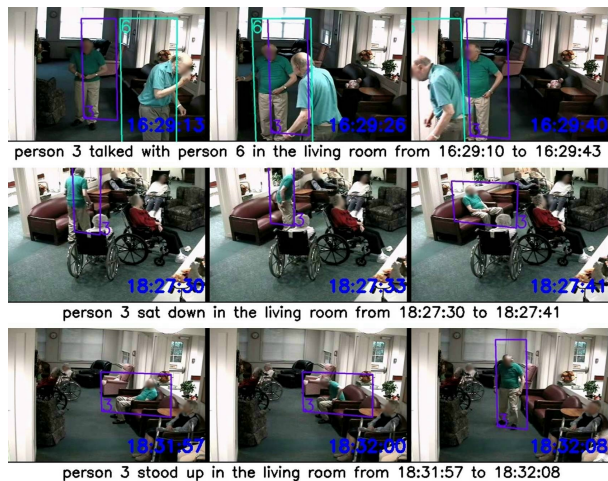
We performed long-term surveillance video summarization experiments by generating visual diaries on the *Caremedia Long* sequence. To acquire ground truth for activity detection experiments, we manually labeled the activities of three residents throughout the sequence. The nursing home residents were selected because they are the people we would like to focus on for the automatic analysis of health status.

We evaluated the different aspects of the visual diary: “room/state timing estimates”, “interaction timing estimates”, “interacting target prediction” and “snippet generation”.

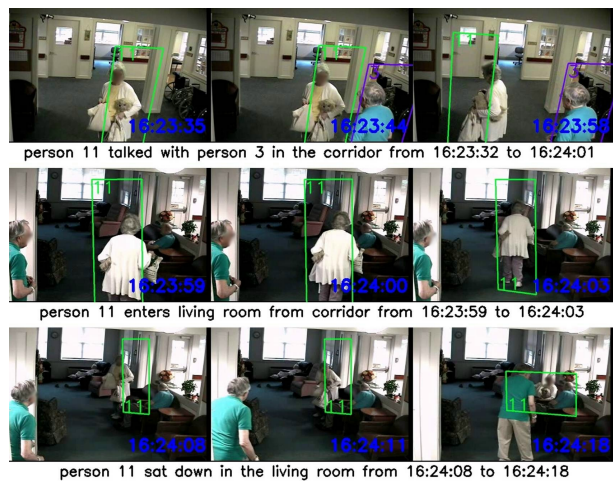
The evaluation of “room/state timing estimates”, i.e. predicted room location and state (sitting or upright), of a person was done on the video frame level. A frame was counted as true positive if the predicted state for a given video frame agrees with the ground truth. False positives and false negatives were computed similarly.

To evaluate “interaction timing estimates”, i.e. how much time a person spent in interactions, a frame was only counted as true positive if 1) both the prediction result and ground truth result agree that there was interaction and 2) the ID of the interacting targets match. False positives and false negatives were computed similarly. For “interacting target prediction”, i.e. who interacted with whom, a true positive was counted when the predicted and ground truth output both agree that the resident interacted with a given person. False positives and false negatives were computed similarly.

The evaluation of “snippet generation” accuracy was done as follows. For snippets related to sit down, stand up and room change activities, a snippet was correct if the predicted result and ground truth result had less than a 5 second time difference. For social interaction-related snippets, a snippet was correct if more than 50% of the predicted snippet contained a matching ground truth interaction. Also, if a ground truth interaction was predicted as three separate interactions, then only one interaction was counted as true positive while the other two were counted as false positives. This prevents double counting of a single ground-truth interaction.



(a) Example snippets for resident 3.



(b) Example snippets for resident 11.

Fig. 8: Example visual diary snippets for each resident.

Based on the tracking output, we performed activity detection and visual diary generation on the three residents. 184 ground-truth snippets were annotated. The performance of visual diary generation is summarized in Table 2. From the table, 38% of the generated snippets were correct, and we have successfully retrieved 52% of the activities-of-interest. For “room/state timing estimates”, a 51.1% recall shows that we know the state and room location of a person more than 50% of the time. The lower performance for “interaction timing estimates” was mainly caused by tracking failures, as both persons need to be tracked correctly for interactions to be correctly detected and timings to be accurate. However, if we only want to know the interaction targets, we still can achieve 63% F1-score. These numbers are not high, but given that our method is fully automatic other than the collection of the face gallery, this is a first cut at generating visual diaries for the elderly by summarizing hundreds or even thousands of hours of surveillance video.

As our visual diary generation is heavily based on tracking, we analyze the effect of tracking performance on visual diary generation accuracy. We computed the snippet generation F1-score for multiple tracking runs with varying tracking performance. These runs include our baseline runs and also runs where we randomly corrupted face recognition labels to decrease tracking performance. Results are shown in Figure 9, which shows that as tracking performance increases, snippet generation F1 also increases with a trend which could be fitted by a second-order polynomial.

Figure 8 shows example visual diary snippets for residents ID 3 and 11. From the generated snippets, we can clearly see what each resident was doing at each time of the day. Long term statistics were also compiled as shown in Figure 2, which can clearly show the amount of time spent by each person in each room and in social interactions. If these statistics were computed over many days, a doctor or staff member could start looking for patterns to assess the status of health of a resident. Compared to tedious manual analysis of hundreds of hours of surveillance video, our method offers a strong alternative for

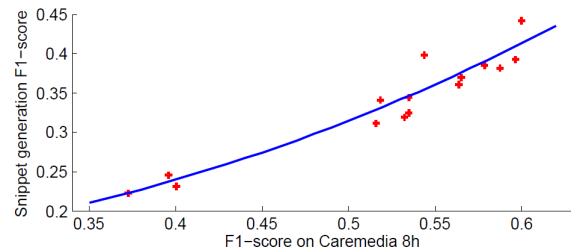


Fig. 9: Performance of snippet generation under varying tracking performance.

the analysis of long-term surveillance video.

5 CONCLUSION

We proposed a tracking algorithm which utilizes identity information such as face recognition to not only enhance multi-person tracking performance, but also assign a real-world identity to each tracked target. Tracking experiments performed on up to 116.25 hours of video in a complex indoor environment showed that our tracker is an improvement over the state-of-the-art in long-term identity-aware multi-person tracking. Also, we proposed to generate visual diaries with a summarization-by-tracking framework for identity-aware video summarization. Experiments performed on 116.25 hours of video showed that we can generate visual diary snippets with 38% precision and 52% recall. Though our numbers are not high, compared to tedious manual analysis of hundreds of hours of surveillance video, our method is a strong alternative for the analysis of long-term surveillance video, and it potentially opens the door to summarization of hundreds or even thousands of hours of surveillance video.

REFERENCES

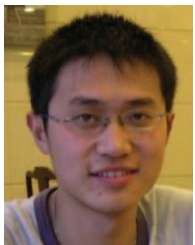
- [1] L. Zhang, Y. Li, and R. Nevatia, “Global data association for multi-object tracking using network flows,” in *CVPR*, 2008.
- [2] R. T. Collins, “Multitarget data association with higher-order motion models,” in *CVPR*, 2012.

- [3] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *CVPR*, 2012.
- [4] K. Okuma, A. Taleghani, N. D. Freitas, O. D. Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *ECCV*, 2004.
- [5] J. K. Rowling, *Harry Potter and the Prisoner of Azkaban*. London: Bloomsbury, 1999.
- [6] S.-I. Yu, Y. Yang, and A. Hauptmann, "Harry Potter's Marauder's Map: Localizing and tracking multiple persons-of-interest by nonnegative discretization," in *CVPR*, 2013.
- [7] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," in *IEEE TPAMI*, 2008.
- [8] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," in *IEEE TPAMI*, 2011.
- [9] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Multi-commodity network flow for tracking multiple people," in *IEEE TPAMI*, 2014.
- [10] X. Wang, E. Turetken, F. Fleuret, and P. Fua, "Tracking interacting objects optimally using integer programming," in *ECCV*, 2014.
- [11] A. R. Zamir, A. Dehghan, and M. Shah, "GMCP-tracker: global multi-object tracking using generalized minimum clique graphs," in *ECCV*, 2012.
- [12] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *CVPR*, 2011.
- [13] A. Andriyenko and K. Schindler, "Multi-target tracking by continuous energy minimization," in *CVPR*, 2011.
- [14] A. Milan, K. Schindler, and S. Roth, "Detection-and trajectory-level exclusion in multiple object tracking," in *CVPR*, 2013.
- [15] A. Butt and R. Collins, "Multi-target tracking by Lagrangian relaxation to min-cost network flow," in *CVPR*, 2013.
- [16] C. Huang, B. Wu, and R. Nevatia, "Robust object tracking by hierarchical association of detection responses," in *ECCV*, 2008.
- [17] C.-H. Kuo, C. Huang, and R. Nevatia, "Multi-target tracking by on-line learned discriminative appearance models," in *CVPR*, 2010.
- [18] C.-H. Kuo and R. Nevatia, "How does person identity recognition help multi-person tracking?" in *CVPR*, 2011.
- [19] B. Yang and R. Nevatia, "An online learned CRF model for multi-target tracking," in *CVPR*, 2012.
- [20] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: Hybridboosted multi-target tracker for crowded scene," in *CVPR*, 2009.
- [21] B. Yang and R. Nevatia, "Multi-target tracking by online learning of non-linear motion patterns and robust appearance models," in *CVPR*, 2012.
- [22] H. Jiang, S. Fels, and J. J. Little, "A linear programming approach for multiple object tracking," in *CVPR*, 2007.
- [23] B. Leibe, K. Schindler, and L. Van Gool, "Coupled detection and trajectory estimation for multi-object tracking," in *CVPR*, 2007.
- [24] A. Andriyenko and K. Schindler, "Globally optimal multi-target tracking on a hexagonal lattice," in *ECCV*, 2010.
- [25] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.
- [26] S.-H. Bae and K.-J. Yoon, "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in *CVPR*, 2014.
- [27] B. Wang, G. Wang, K. L. Chan, and L. Wang, "Tracklet association with online target-specific metric learning," in *CVPR*, 2014.
- [28] X. Zhang, W. Hu, S. Maybank, and X. Li, "Graph based discriminative learning for robust and efficient object tracking," in *CVPR*, 2007.
- [29] W. Hu, X. Li, W. Luo, X. Zhang, S. Maybank, and Z. Zhang, "Single and multiple object tracking using log-Euclidean Riemannian subspace and block-division appearance model," *IEEE TPAMI*, 2012.
- [30] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE TPAMI*, 2005.
- [31] R. Hess and A. Fern, "Discriminatively trained particle filters for complex multi-object tracking," in *CVPR*, 2009.
- [32] C. Dicle, M. Sznajder, and O. Camps, "The way they move: Tracking targets with similar appearance," in *ICCV*, 2013.
- [33] A. A. Butt and R. T. Collins, "Multiple target tracking using frame triplets," in *ACCV*, 2013.
- [34] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Real-time affine region tracking and coplanar grouping," in *CVPR*, 2001.
- [35] M. J. Marín-Jiménez, A. Zisserman, M. Eichner, and V. Ferrari, "Detecting people looking at each other in videos," *IJCV*, 2014.
- [36] S. Gold, A. Rangarajan *et al.*, "Softmax to softassign: Neural network algorithms for combinatorial optimization," *Journal of Artificial Neural Networks*, 1996.
- [37] V. Chari, S. Lacoste-Julien, I. Laptev, and J. Sivic, "On pairwise costs for network flow multi-object tracking," in *CVPR*, 2015.
- [38] M. Zervos, H. BenShitrit, F. Fleuret, and P. Fua, "Facial descriptors for identity-preserving multiple people tracking," Technical Report EPFL-ARTICLE-187534, 2013.
- [39] A. Dehghan, Y. Tian, P. H. Torr, and M. Shah, "Target identity-aware network flow for online multiple target tracking," in *CVPR*, 2015.
- [40] M. Everingham, J. Sivic, and A. Zisserman, "Taking the bite out of automated naming of characters in TV video," *Image and Vision Computing*, 2009.
- [41] J. Sivic, M. Everingham, and A. Zisserman, "Who are you? Learning person specific classifiers from video," in *CVPR*, 2009.
- [42] A. Y. Ng, M. I. Jordan, Y. Weiss *et al.*, "On spectral clustering: Analysis and an algorithm," *NIPS*, 2002.
- [43] C. Lu and X. Tang, "Surpassing human-level face verification performance on LFW with GaussianFace," *arXiv preprint arXiv:1404.3840*, 2014.
- [44] Y. Yang, H. T. Shen, F. Nie, R. Ji, and X. Zhou, "Nonnegative spectral clustering with discriminative regularization," in *AAAI*, 2011.
- [45] Z. Yang and E. Oja, "Linear and nonlinear projective nonnegative matrix factorization," in *Neural Networks, IEEE Transactions on*, 2010.
- [46] C. Ding, T. Li, and M. I. Jordan, "Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding," in *ICDM*, 2008.
- [47] J. Yoo and S. Choi, "Nonnegative matrix factorization with orthogonality constraints," *Journal of Computing Science and Engineering*, 2010.
- [48] F. Pompili, N. Gillis, P.-A. Absil, and F. Glineur, "Two algorithms for orthogonal nonnegative matrix factorization with application to clustering," *Neurocomputing*, 2014.
- [49] S. J. Wright and J. Nocedal, *Numerical Optimization*. Springer, New York, 1999, vol. 2.
- [50] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural computation*, 2007.
- [51] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, 2000.
- [52] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, and Y. Pan, "A multimedia retrieval framework based on semi-supervised ranking and relevance feedback," in *IEEE TPAMI*, 2012.
- [53] A. Ellis, A. Shahrokni, and J. Ferryman, "PETS2009 and winter-PETS 2009 results: A combined evaluation," in *Performance Evaluation of Tracking and Surveillance (PETS-Winter)*, 2009.
- [54] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR*, 2011.
- [55] "National institute of standards and technology: TRECVID 2012 evaluation for surveillance event detection. <http://www.nist.gov/speech/tests/trecvid/2012/>," 2012.
- [56] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," in *CVPR*, 2011.
- [57] Y. Yang, A. Hauptmann, M.-Y. Chen, Y. Cai, A. Bharucha, and H. Wactlar, "Learning to predict health status of geriatric patients from observational data," in *Computational Intelligence in Bioinformatics and Computational Biology*, 2012.
- [58] "Gurobi optimizer reference manual, <http://www.gurobi.com/>," 2012.
- [59] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," in *Neural computation*, 2003.
- [60] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," in *IEEE TPAMI*, 2010.
- [61] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester, "Discriminatively trained deformable part models, release 5," <http://people.cs.uchicago.edu/~rbg/latent-release5/>.
- [62] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.
- [63] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *CVPR*, 1999.
- [64] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," in *J. Image Video Process.*, 2008.
- [65] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol," *IEEE TPAMI*, 2009.
- [66] M. A. Sadeghi and D. Forsyth, "30hz object detection with DPM V5," 2014.

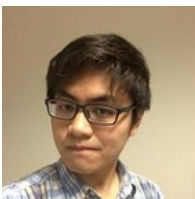
- [67] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian computer vision system for modeling human interactions," in *IEEE TPAMI*, 2000.
- [68] C. McPhail and R. T. Wohlstein, "Using film to analyze pedestrian behavior," in *Sociological Methods & Research*, 1982.
- [69] H. Wang, C. Schmid *et al.*, "Action recognition with improved trajectories," in *ICCV*, 2013.
- [70] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: An evaluation of recent feature encoding methods," in *BMVC*, 2011.



Shoou-I Yu Shoou-I Yu received the B.S. in Computer Science and Information Engineering from National Taiwan University, Taiwan in 2009. He is now a Ph.D. student in Language Technologies Institute, Carnegie Mellon University. His research interests include multi-object tracking and multimedia retrieval.



Yi Yang Yi Yang received the PhD degree from Zhejiang University in 2010. He was a postdoc research fellow with the School of Computer Science at Carnegie Mellon University. He is now an Associate Professor with University of Technology Sydney. His research interest include multimedia, computer vision and machine learning.



Xuanchong Li Xuanchong Li received B.E. in computer science and technology from Zhejiang University, China in 2012. He is now a master student in Carnegie Mellon University. His research interest includes computer vision, machine learning.



Alexander G. Hauptmann Alexander G. Hauptmann received the B.A. and M.A. degrees in psychology from The Johns Hopkins University, Baltimore, MD, USA, in 1982, the "Diplom" in computer science from the Technische Universität Berlin, Berlin, Germany, in 1984, and the Ph.D. degree in computer science from Carnegie Mellon University (CMU), Pittsburgh, PA, USA in 1991. He is a Principal Systems Scientist in the CMU Computer Science Department and also a faculty member with CMU's

Language Technologies Institute. His research combines the areas of multimedia analysis and retrieval, man-machine interfaces, language processing, and machine learning. He is currently leading the Informedia project which engages in understanding of video data ranging from news to surveillance, Internet video for applications in general retrieval as well as healthcare.